

CLAIM AMENDMENT(S)

1. (original) One or more electronically-accessible media comprising electronically-executable instructions that comprise at least part of an operating system, the at least part of the operating system including:

an application programming interface that is capable of creating callback-type dynamic function tables, each callback-type dynamic function table including a begin address, an end address, and a callback function; wherein the operating system calls the callback function when an exception occurs within an address range bounded by the begin address and the end address, the callback function requesting exception handling and/or unwind information from a runtime component.

2. (original) The one or more electronically-accessible media as recited in claim 1, wherein the callback function returns from the runtime component data for code address-to-pointer information.

3. (original) The one or more electronically-accessible media as recited in claim 2, wherein the data for code address-to-pointer information comprises a reference to the code address-to-pointer information.

1 4. (original) The one or more electronically-accessible media as recited
2 in claim 2, wherein the data for code address-to-pointer information comprises the
3 code address-to-pointer information.

4

5 5. (original) The one or more electronically-accessible media as recited
6 in claim 4, wherein the code address-to-pointer information comprises a start
7 address, a final address, and an unwind pointer.

8

9 6. (original) One or more electronically-accessible media comprising
10 electronically-executable instructions that utilize an application programming
11 interface, the application programming interface facilitating creation of callback-
12 type dynamic function tables; each callback-type dynamic function table including
13 a begin address, an end address, and a callback function, each callback-type
14 dynamic function table corresponding to a code heap that stores code for a
15 plurality of functions in a runtime environment; wherein interaction between the
16 runtime environment and an operating system is precipitated upon calling the
17 callback function to acquire exception handling and/or unwind information.

18

19 7. (original) The one or more electronically-accessible media as recited
20 in claim 6, wherein the interaction between the runtime environment and the
21 operating system comprises (i) the operating system requesting exception handling
22 and/or unwind information from the runtime environment and (ii) the runtime
23 environment providing the exception handling and/or unwind information to the
24 operating system.

1 8. (original) The one or more electronically-accessible media as recited
2 in claim 6, wherein the begin address and the end address of each callback-type
3 dynamic function table comprises a begin address and an end address of a
4 respective code heap to which a respective callback-type dynamic function table
5 corresponds.

6

7 9. (original) The one or more electronically-accessible media as recited
8 in claim 6, wherein the exception handling and/or unwind information comprises
9 data for code address-to-pointer information for a particular function of the
10 plurality of functions.

11

12 10. (original) The one or more electronically-accessible media as recited
13 in claim 9, wherein the particular function of the plurality of functions is
14 ascertained using an instruction pointer that is passed as an argument of the
15 callback function.

16

17 11. (original) The one or more electronically-accessible media as recited
18 in claim 6, wherein each callback-type dynamic function table is created, at least in
19 part, by adding each callback-type dynamic function table to a linked list of
20 dynamic function tables.

21

22 12. (original) The one or more electronically-accessible media as recited
23 in claim 11, wherein the linked list of dynamic function tables includes one or
24 more non-callback-type dynamic function tables.

1 **13.** (original) One or more electronically-accessible media comprising
2 electronically-executable instructions that comprise at least part of a runtime
3 component, the runtime component adapted to manage exception handling and/or
4 unwind information for a plurality of functions that are executing as part of a
5 runtime environment, the runtime component capable of providing data for code
6 address-to-pointer information to an operating system when an exception occurs
7 with respect to a particular function of the plurality of functions.

8

9 **14.** (original) The one or more electronically-accessible media as recited
10 in claim 13, wherein the data for code address-to-pointer information comprises a
11 reference to the code address-to-pointer information for the particular function of
12 the plurality of functions.

13

14 **15.** (original) The one or more electronically-accessible media as recited
15 in claim 13, wherein the data for code address-to-pointer information comprises
16 the code address-to-pointer information for the particular function of the plurality
17 of functions.

18

19 **16.** (original) The one or more electronically-accessible media as recited
20 in claim 15, wherein the code address-to-pointer information comprises a start
21 address, a final address, and an unwind pointer for the particular function of the
22 plurality of functions.

1 **17.** (original) An electronic device comprising:

2 a runtime environment that is managing code for a plurality of functions;

3 and

4 an operating system that is managing a linked list of dynamic function
5 tables that are searched when an exception occurs, the operating system adapted to
6 call a callback function as indicated by a dynamic function table of the linked list
7 of dynamic function tables to request that the runtime environment provide
8 exception handling and/or unwind information for at least one function of the
9 plurality of functions;

10 wherein the runtime environment is capable of providing to the operating
11 system the exception handling and/or unwind information for the at least one
12 function of the plurality of functions responsive to the callback function.

13

14 **18.** (original) The electronic device as recited in claim 17, wherein the
15 dynamic function table comprises a callback dynamic function table.

16

17 **19.** (original) The electronic device as recited in claim 18, wherein the
18 linked list of dynamic function tables comprises at least one sorted dynamic
19 function table and at least one unsorted dynamic function table.

20

21 **20.** (original) The electronic device as recited in claim 17, wherein the
22 dynamic function table corresponds to more than one function of the plurality of
23 functions.

1 **21.** (original) The electronic device as recited in claim 17, wherein the
2 dynamic function table includes a begin address entry and an end address entry that
3 reflect a begin address and an end address, respectively, of a code heap that
4 includes more than one function of the plurality of functions, including the at least
5 one function of the plurality of functions.

6

7 **22.** (original) The electronic device as recited in claim 17, wherein the
8 runtime environment comprises at least one just-in-time (JIT) compiler.

9

10 **23.** (original) The electronic device as recited in claim 17, wherein the
11 exception handling and/or unwind information comprises data for code address-to-
12 pointer information for the at least one function of the plurality of functions.

13

14 **24.** (original) The electronic device as recited in claim 23, wherein the
15 data for code address-to-pointer information for the at least one function of the
16 plurality of functions directly or indirectly provides to the operating system a start
17 address, a final address, and an unwind pointer for the at least one function of the
18 plurality of functions.

1 **25.** (original) One or more electronically-accessible media comprising a
2 data structure, the data structure comprising:

3 a begin address;
4 an end address; and
5 a callback function that, when called, returns from a runtime environment
6 exception handling and/or unwind information for a function associated with at
7 least one address that is between the begin address and the end address.

8

9 **26.** (original) The one or more electronically-accessible media as recited
10 in claim 25, wherein the begin address and the end address reflect a begin address
11 and an end address, respectively, for a code heap of the runtime environment.

12

13 **27.** (original) The one or more electronically-accessible media as recited
14 in claim 26, wherein the code heap includes code for a plurality of functions.

15

16 **28.** (original) The one or more electronically-accessible media as recited
17 in claim 27, wherein the callback function, when called, is capable of returning
18 from the runtime environment exception handling and/or unwind information for
19 any function of the plurality of functions.

20

21 **29.** (original) The one or more electronically-accessible media as recited
22 in claim 25, wherein the callback function is called by an operating system to
23 enable the operating system to unwind a stack.

1 30. (original) The one or more electronically-accessible media as recited
2 in claim 25, wherein the callback function accepts as input the at least one address
3 that is associated with the function, the at least one address comprising an
4 instruction pointer.

5

6 31. (original) The one or more electronically-accessible media as recited
7 in claim 25, wherein the callback function accepts as input a reference to at least
8 part of the data structure.

9

10 32. (original) The one or more electronically-accessible media as recited
11 in claim 25, wherein the exception handling and/or unwind information comprises
12 data for code address-to-pointer information for the function that is associated with
13 the at least one address.

14

15 33. (original) The one or more electronically-accessible media as recited
16 in claim 25, wherein the begin address, the end address, and the callback function
17 together comprise a callback dynamic function table.

18

19 34. (original) The one or more electronically-accessible media as recited
20 in claim 33, wherein the data structure further comprises:

21 a plurality of dynamic function tables, the plurality of dynamic function
22 tables including the callback dynamic function table.

1 **35.** (original) An electronic device comprising:
2 at least one processor; and
3 one or more media in operative communication with the at least one
4 processor, the one or more media including a data structure comprising:
5 a begin address value;
6 an end address value; and
7 a callback function that, when called, returns from a runtime
8 environment exception handling and/or unwind information for a function
9 associated with at least one address that is between the begin address value
10 and the end address value.

11
12 **36.** (original) The electronic device as recited in claim 35, wherein the
13 one or more media further include a code heap managed by the runtime
14 environment, the code heap having a begin address and an end address; the begin
15 address value and the end address value reflecting the begin address and the end
16 address, respectively, of the code heap.

17
18 **37.** (original) The electronic device as recited in claim 36, wherein the
19 code heap includes code for a plurality of functions.

20
21 **38.** (original) The electronic device as recited in claim 37, wherein the
22 callback function, when called, is capable of returning from the runtime
23 environment exception handling and/or unwind information for any function of the
24 plurality of functions responsive to the at least one address.

1 **39.** (original) The electronic device as recited in claim 35, wherein the
2 one or more media further include a stack; and wherein the callback function is
3 called by an operating system to enable the operating system to unwind the stack.

4

5 **40.** (original) One or more electronically-accessible media comprising
6 electronically-executable instructions that include:

7 a callback function, the callback function accepting as input an instruction
8 pointer that is associated with an address of a function from a runtime environment
9 and producing as output data for code address-to-pointer information for the
10 function having the address that is associated with the instruction pointer;

11 wherein the callback function may be called by an operating system and
12 implemented by the runtime environment.

13

14 **41.** (original) The one or more electronically-accessible media as recited
15 in claim 40, wherein the callback function further accepts as input a reference to a
16 callback dynamic function table for context.

17

18 **42.** (original) The one or more electronically-accessible media as recited
19 in claim 41, wherein the callback dynamic function table includes a begin address
20 and an end address that define an address range that includes the instruction
21 pointer.

1 **43.** (original) The one or more electronically-accessible media as recited
2 in claim 40, wherein the data for code address-to-pointer information that is output
3 by the callback function is provided from the runtime environment to the operating
4 system.

5

6 **44.** (original) The one or more electronically-accessible media as recited
7 in claim 40, wherein the data for code address-to-pointer information comprises a
8 reference to the code address-to-pointer information.

9

10 **45.** (original) The one or more electronically-accessible media as recited
11 in claim 44, wherein the referenced code address-to-pointer information comprises
12 a start address, a final address, and an unwind pointer for the function having the
13 address that is associated with the instruction pointer.

14

15 **46.** (original) The one or more electronically-accessible media as recited
16 in claim 40, wherein the data for code address-to-pointer information comprises
17 the code address-to-pointer information.

18

19 **47.** (original) The one or more electronically-accessible media as recited
20 in claim 40, wherein the callback function is adapted to be called by the operating
21 system during an exception handling procedure.

1 **48.** (original) The one or more electronically-accessible media as recited
2 in claim 40, wherein the callback function is implemented, at least partially, by the
3 runtime environment by inspecting a code header for the function having the
4 address that is associated with the instruction pointer.

5
6 **49.** (original) The one or more electronically-accessible media as recited
7 in claim 48, wherein the data for code address-to-pointer information is derived
8 using the code header.

9
10 **50.** (original) The one or more electronically-accessible media as recited
11 in claim 48, wherein the code header is ascertained using a heap structure contents
12 of a code heap that includes code for the function having the address that is
13 associated with the instruction pointer.

14
15 **51.** (original) One or more electronically-accessible media comprising at
16 least part of an operating system that is configured to request from a runtime
17 environment exception handling and/or unwinding information for functions that
18 are managed by the runtime environment.

19
20 **52.** (original) The one or more electronically-accessible media as recited
21 in claim 51, wherein the at least part of the operating system is further configured
22 to request from the runtime environment the exception handling and/or unwinding
23 information after an exception is discovered.

1 **53.** (original) The one or more electronically-accessible media as recited
2 in claim 51, wherein the at least part of the operating system is further configured
3 to request the exception handling and/or unwinding information from the runtime
4 environment responsive to locating a dynamic function table having a callback
5 function.

6

7 **54.** (original) The one or more electronically-accessible media as recited
8 in claim 53, wherein the dynamic function table having the callback function
9 corresponds to at least two functions compiled in the runtime environment by a
10 just-in-time (JIT) compiler.

11

12 **55. - 61.** (canceled)

13

14

15

16

17

18

19

20

21

22

23

24

25

1 **62.** (original) One or more electronically-accessible media comprising
2 electronically-executable instructions that, when executed, direct an electronic
3 device to perform actions comprising:

4 receiving at a runtime an instruction pointer with a request for data for code
5 address-to-pointer information;

6 ascertaining a runtime function associated with the instruction pointer;

7 accessing a code header of the ascertained runtime function to extract data
8 for code address-to-pointer information for the ascertained runtime function; and

9 providing the extracted data for code address-to-pointer information from
10 the runtime.

11
12 **63.** (original) The one or more electronically-accessible media
13 comprising the electronically-executable instructions that, when executed, direct
14 an electronic device to perform the actions as recited in claim 62, wherein the
15 action of providing comprises the action of:

16 providing from the runtime a pointer to the code address-to-pointer
17 information for the ascertained runtime function.

18
19 **64.** (original) The one or more electronically-accessible media
20 comprising the electronically-executable instructions that, when executed, direct
21 an electronic device to perform the actions as recited in claim 62, wherein the
22 action of providing comprises the action of:

23 providing from the runtime to an operating system the extracted data
24 for code address-to-pointer information.

1 **65.** (original) An electronic device comprising:

2 at least one processor; and

3 one or more media including processor-executable instructions that are
4 capable of being executed by the at least one processor, the processor-executable
5 instructions adapted to cause the electronic device to perform actions comprising:

6 locating a callback dynamic function table having a callback
7 function;

8 initiating, by an operating system, the callback function to interact
9 with a runtime;

10 receiving data for code address-to-pointer information from the
11 runtime responsive to the initiating; and

12 using the received data for code address-to-pointer information to
13 attain information for at least one of exception handling and stack
14 unwinding.

15
16 **66.** (original) The electronic device as recited in claim 65, wherein:

17 the receiving action comprises:

18 receiving a reference to code address-to-pointer information
19 from the runtime; and

20 the using action comprises:

21 using the reference to attain code address-to-pointer
22 information, including an unwind pointer; and

23 using the unwind pointer to attain unwind information.

1 67. (original) The electronic device as recited in claim 65, wherein the
2 processor-executable instructions are adapted to cause the electronic device to
3 perform a further action comprising:

4 discovering an exception from a runtime function.

5

6 68. (original) The electronic device as recited in claim 65, wherein the
7 processor-executable instructions are adapted to cause the electronic device to
8 perform a further action comprising:

9 searching a dynamic function table linked list using an instruction
10 pointer;

11 wherein the locating action comprises:

12 locating the callback dynamic function table having the
13 callback function from the dynamic function table linked list using
14 the instruction pointer, the callback dynamic function table including
15 a begin address that is less than and an end address that is greater
16 than the instruction pointer.

1 **69.** (original) The electronic device as recited in claim 68, wherein the
2 processor-executable instructions are adapted to cause the electronic device to
3 perform further actions comprising:

4 receiving at the runtime, responsive to the initiating, the instruction
5 pointer with a request for the data for code address-to-pointer information;

6 ascertaining a runtime function associated with the instruction
7 pointer;

8 accessing a code header of the ascertained runtime function to
9 extract the data for code address-to-pointer information for the ascertained
10 runtime function; and

11 providing, from the runtime to the operating system, the data for
12 code address-to-pointer information.

13
14 **70.** (original) One or more electronically-accessible media comprising
15 electronically-executable instructions that, when executed, direct an electronic
16 device to perform actions comprising:

17 locating a callback dynamic function table having a callback function;

18 initiating, by an operating system, the callback function to interact with a
19 runtime;

20 receiving, at the operating system, data for code address-to-pointer
21 information from the runtime responsive to the initiating; and

22 using the received data for code address-to-pointer information to attain
23 information for at least one of exception handling and stack unwinding.

1 **71.** (canceled)

2

3

4 **72.** (original) An arrangement comprising:

5 means for locating a dynamic function table having a callback function;

6 means for initiating the callback function to interact with a runtime

7 environment;

8 means for receiving data for code address-to-pointer information from the

9 runtime environment responsive to the callback function initiation; and

10 means for using the received data for code address-to-pointer information to

11 attain information for at least one of exception handling and stack unwinding.

12

13

14

15

16

17

18

19

20

21

22

23

24

25

ELECTION OF GROUP

From among the identified Groups as stipulated by the Office Action,

Applicants hereby elect Group I, corresponding

to at least claims 1-54, 62-70, and 72,

for prosecution in the instant Application.